REMARKS

Claims 1-30 are currently pending in the subject application and are presently under consideration. Claims 16, 25, 27 and 28 have been amended as shown on pp. 3-10 of the Reply. Support for these amendments can be found on p. 17, lines 3-22. In addition, the specification has been amended as indicated on p. 2.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I.   Rejection of Claims 1-30 Under 35 U.S.C. §102(e)

Claims 1-30 stand rejected under 35 U.S.C. §102(e) as being anticipated by Williams (US Patent 6,591,272). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Williams does not anticipate each and every element as set forth in the subject claims.

> A single prior art reference anticipates a patent claim only if it expressly or inherently describes each and every limitation set forth in the patent claim. *Trintec Industries, Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); See *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The identical invention must be shown in as complete detail as is contained in the ... claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

The claimed invention relates to systems and methods for making asynchronous calls using a common pattern. More particularly, independent claim 1 recites a system for *converting a synchronous method call on a target method to an asynchronous method call, comprising: a pattern generator operable to break the synchronous method call into one or more constituent parts; and a pattern data store, operably connected to the pattern generator, the pattern data store adapted to store data associated with converting a synchronous method call to an asynchronous method call.* Williams does not expressly or inherently disclose the aforementioned novel aspects of applicants' invention as recited in the subject claims.

More particularly, Williams does not disclose or suggest a system for *converting a synchronous method call to an asynchronous method call,* as recited in independent claim 1. The Office Action contends that Williams discloses a system for converting synchronous method calls, at col. 15, lines 27-53. (*See* Office Action dated November 1, 2005, page 2). Applicants'

11

representative respectfully disagrees. Williams discloses a method and apparatus for transmitting objects from a database on a server computer to a client computer. Contents of databases are translated into objects by reading the database schema metadata to determine data interrelationships and create objects with nominal human to computer interaction. (*See* Abstract). Further, a comprehensive exception handling scheme handles all server-side exceptions, standardizes and normalizes them then transmits the exceptions via CORBA. A minor CORBA-2 limitation is that exceptions may only be thrown over the network when the synchronous, blocking remote accessors are used. Synchronous remote CORBA methods are available for PRO-OBJECT server implementations, but no exceptions will be thrown over CORBA by default. This is because PRO-OBJECTS by default utilize the asynchronous IIOP accessors. Further, OSF CORBA-style PRO-OBJECTS utilize by default asynchronous IIOP Distributed Callbacks. Early prototypes used standard synchronous method calls and this was not optimal. (*See* Col. 15, lines 27-53).

It would appear that Williams merely discloses the use of synchronous method calls and asynchronous method calls with regard to PRO-OBJECT server implementations. According to Williams, asynchronous method calls are optimal with regard to OSF CORBA-style PRO-OBJECTS. Williams does not disclose a system of *converting* synchronous method calls, but merely states that asynchronous method calls are preferred. Applicants' claimed invention provides for a system of *converting a synchronous method call to an asynchronous method call.*

Furthermore, Williams fails to expressly or inherently disclose *a pattern generator operable to break the synchronous method call into one or more constituent parts* as claimed. The Office Action contends that Williams discloses a pattern generator at the Abstract, lines 1-17 and at Col. 1, lines 23-29. (*See* Office Action dated November 1, 2005, page 2). Applicants' representative respectfully disagrees. As stated above, Williams discloses that the contents of databases are translated into objects by reading the database schema metadata to determine data interrelationships and create objects with nominal human to computer interaction. Metadata for any number of databases is normalized in a standardized view. Data objects are then produced by encapsulating the metadata and data values. (*See* Abstract).

Accordingly, Williams discloses taking the contents of various databases and translating them into objects. The data objects of Williams are produced by encapsulating the metadata and data values from the database schema. Williams is silent in regards to a pattern generator used to break the *synchronous method calls* into constituent parts as claimed. By employing the pattern

12

generator and the data stored in the pattern data store, the present invention mitigates the need for a called object to be reprogrammed for supporting asynchronous behavior by its clients, providing advantages over conventional systems where it is typically the called object that determines whether it will proceed synchronously or asynchronously, as in the cited reference. Thus, Williams fails to disclose or suggest *converting a synchronous method call to an asynchronous method call*, let alone disclosing *a pattern generator operable to break the synchronous method call into one or more constituent parts and a pattern data store for storing data*, as recited in claim 1.

Moreover, Williams fails to disclose or suggest a system for making asynchronous calls on a target method, comprising: *synchronous method call code associated with a client caller, the synchronous method call code is broken into constituent parts; an asynchronous call initializer adapted to accept input parameters from the client caller and to forward the input parameters towards the target method, ...the asynchronous call initializer further adapted to accept a state object and to populate one or more fields in the state object with state values associated with the asynchronous call, the asynchronous call initializer further adapted to return a result object to the client caller; an asynchronous call completer adapted to accept results generated by the target method and to supply the results to the client caller, the asynchronous call completer further adapted to update the state object, the asynchronous call completer further adapted to update the result object; and a state tracker, operable to track and log state related to processing associated with the asynchronous call initializer, the asynchronous call completer and the target method, the state tracker further operable to update the state object*, as recited in independent claim 16 (and similarly independent claims 25, 27 and 28).

Williams relates to translating the contents of databases into objects by reading the database schema metadata to determine data interrelationships and creating objects. (*See* Abstract). Metadata for any number of databases is normalized in a standardized view. Data objects are then produced by encapsulating the metadata and data values. (*See* Abstract). In response to an object access request, the server computer generates a pseudo-object and associated metadata. The pseudo-object is transported as a single network packet unit transmitted synchronously or asynchronously over the network to the client computer. (*See* Col. 8, lines 11-22).

It would appear that Williams discloses taking the contents of various databases and translating them into objects. The data objects of Williams are produced by encapsulating the metadata and data values from the database schema. These objects are then transported as a single

13

network packet unit and transmitted synchronously or asynchronously over the network. Williams is silent in regards to breaking the *synchronous method call code* into constituent parts and utilizing *an asynchronous call initializer to accept input parameters from the client caller* and *an asynchronous call completer to supply the results to the client caller*, as claimed.

Thus, the present invention, by providing the asynchronous call initializer, which accepts the inputs from the client caller, and by providing the call completer, which returns the results expected by the client caller, facilitates adding asynchronous method call processing to systems that conventionally only have synchronous method call processing available, without requiring the programmer of the client caller or the programmer of the target method to change their code. Accordingly, Williams fails to teach or suggest a system for facilitating asynchronous calls, comprising: ...*synchronous method call code broken into constituent parts; an asynchronous call initializer adapted to accept input parameters from the client caller...; an asynchronous call completer adapted to accept results generated by the target method and to supply the results to the client caller...*

Moreover, Williams fails to disclose or suggest a method for *converting code for a synchronous method call on a target method to code for asynchronous method call, comprising: receiving a code for a synchronous method call; passing the code for the synchronous method call through a call conversion process...; creating an asynchronous call result object to store results associated with the asynchronous method call; and creating an asynchronous call state object to store state information associated with the asynchronous method call*, as recited in independent claim 18 (and similarly independent claims 24, 29 and 30).

Williams relates to translating the contents of databases into objects by reading the database schema metadata to determine data interrelationships and creating objects. (*See* Abstract). In response to an object access request, the server computer generates a pseudo-object and associated metadata. The pseudo-object is transported as a single network packet unit transmitted synchronously or asynchronously over the network to the client computer. (*See* Col. 8, lines 11-22). Further, a comprehensive exception handling scheme handles all server-side exceptions, standardizes and normalizes them then transmits the exceptions via CORBA. Synchronous remote CORBA methods are available for PRO-OBJECT server implementations, but no exceptions will be thrown over CORBA by default. This is because PRO-OBJECTS by default utilize the asynchronous IIOP accessors. (*See* Col. 15, lines 27-53). Thus, it would appear that Williams

14

merely discloses the use of synchronous method calls and asynchronous method calls with regard to PRO-OBJECT server implementations. Williams does not disclose utilizing a call conversion process for *converting* synchronous method calls to asynchronous method calls. Accordingly, Williams fails to teach or suggest a method for *receiving code from a synchronous method call, passing the code through a call conversion process; and converting the code to an asynchronous method call* as claimed.

In view of at least the above, it is readily apparent that Williams fails to expressly or inherently disclose applicants' claimed invention as recited in independent claims 1, 16, 18, 24, 25, 27, 28, 29 and 30 (and claims 2-15, 17, 19-23 and 26 which respectively depend there from). Accordingly, it is respectfully requested that these claims be deemed allowable.
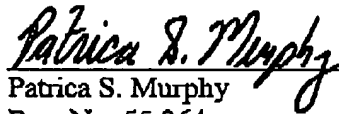
CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP249US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN & TUROCY, LLP

Patrica S. Murphy
Reg. No. 55,964

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731

16